3F8 Inference
by Ziyi Z., Page 1 in 3

1 Regression

1.1 Linear Regression

For a data set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ formed by pairs of input vectors $\mathbf{x}_n = (x_{n,1}, \ldots, x_{n,D})^\top$ and outputs $y_n \in \mathbb{R}$, we assume:

$$y_n = [w_0, \ldots, w_D] \begin{bmatrix} 1 \\ x_{n,1} \\ \vdots \\ x_{n,D} \end{bmatrix} + \epsilon_n$$
$$= \mathbf{w}^\top \widetilde{\mathbf{x}}_n + \epsilon_n$$

The errors or noise $\epsilon_n \sim \mathcal{N}\left(0, \sigma^2\right)$ and $\widetilde{\mathbf{x}}_n = (1, \mathbf{x}_n)^\top, \boldsymbol{\theta} = \left\{\sigma^2, \mathbf{w}\right\}$.

$$p\left(y_n \mid \widetilde{\mathbf{x}}_n, \boldsymbol{\theta}\right) = \mathcal{N}\left(y_n \mid \mathbf{w}^\top \widetilde{\mathbf{x}}_n, \sigma^2\right)$$

$\mathbf{w}$ are the coefficients, weights, etc. (w_0 is called the bias or intercept).

1.2 Maximum Likelihood Estimate

Maximize the *likelihood* function $p\left(y_1, \ldots, y_n \mid \widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_n, \boldsymbol{\theta}\right)$ with respect to $\boldsymbol{\theta}$.
The log-likelihood function:

$$\mathcal{L}(\theta) = \log \prod_{n=1}^N p\left(y_n \mid \widetilde{\mathbf{x}}_n, \boldsymbol{\theta}\right)$$
$$= \sum_{n=1}^N \log \mathcal{N}\left(y_n \mid \mathbf{w}^\top \widetilde{\mathbf{x}}_n, \sigma^2\right)$$

Let $\mathbf{y} = (y_1, \ldots, y_n)^\top, \widetilde{\mathbf{X}} = (\widetilde{\mathbf{x}}_1; \ldots; \widetilde{\mathbf{x}}_n)^\top$:

$$\mathcal{L}(\theta) = -\frac{N}{2} \log\left(2\pi\sigma^2\right)$$
$$- \frac{\mathbf{y}^\top \mathbf{y}}{2\sigma^2} - \frac{\mathbf{w}^\top \widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}} \mathbf{w}}{2\sigma^2} + \frac{\mathbf{y}^\top \widetilde{\mathbf{X}} \mathbf{w}}{\sigma^2}$$

The *Linear Least Squares Solution* (LLSS):

$$\mathbf{w} = \left(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}}\right)^{-1} \widetilde{\mathbf{X}}^\top \mathbf{y}$$

$$\sigma^2 = \frac{1}{N} (\mathbf{y} - \widetilde{\mathbf{X}}\mathbf{w})^\top (\mathbf{y} - \widetilde{\mathbf{X}}\mathbf{w})$$

Linear regression can model non-linear relationships by replacing $\mathbf{x}$ with some non-linear function of the inputs $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^\top$.

2 Bayesian Linear Regression

2.1 Overfitting

A large number of basis functions can lead to *over-fitting*: the model fits the *training data* well but it performs poorly on new *test data*.

2.2 Bayesian Inference

Assume a *prior distribution* $p(\mathbf{w})$ on the model coefficients. The *posterior distribution* for $\mathbf{w}$ given $\mathcal{D}$ is obtained by Bayes rule:

$$p(\mathbf{w} \mid \mathbf{y}, \widetilde{\mathbf{X}}) = \frac{p(\mathbf{y} \mid \widetilde{\mathbf{X}}, \mathbf{w}) p(\mathbf{w})}{p(\mathbf{y} \mid \widetilde{\mathbf{X}})}$$

The *predictive distribution* for $y_\star$ given a new corresponding $\mathbf{x}_\star$ is:

$$p\left(y_\star \mid \widetilde{\mathbf{x}}_\star, \mathbf{y}, \widetilde{\mathbf{X}}\right) =$$
$$\int p\left(y_\star \mid \widetilde{\mathbf{x}}_\star, \mathbf{w}\right) p(\mathbf{w} \mid \mathbf{y}, \widetilde{\mathbf{X}}) d\mathbf{w}$$

2.3 Multivariate Gaussian Distribution

The density of a D -dimensional vector $\mathbf{x}$ is:

$$\mathcal{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{V}) = \frac{1}{\sqrt{(2\pi)^D |\mathbf{V}|}}$$
$$\exp\left\{-\frac{1}{2}(\mathbf{x} - \mathbf{m})^\top \mathbf{V}^{-1} (\mathbf{x} - \mathbf{m})\right\}$$

The density is proportional to the exponential of a quadratic function of $\mathbf{x}$:

$$p(\mathbf{x}) \propto \exp\left\{-\frac{1}{2} \mathbf{x}^\top \mathbf{V}^{-1} \mathbf{x} + \mathbf{m}^\top \mathbf{V}^{-1} \mathbf{x}\right\}$$

The parameter $\mathbf{m}$ determines *mode location* and $\mathbf{V}$ *scales and rotates* the space.

2.4 Linear Combination of Gaussian Random Variables

Let $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{V}_1)$ and $p(\mathbf{e}) = \mathcal{N}(\mathbf{e} \mid \mathbf{0}, \mathbf{V}_2)$ and assume that, for a matrix $\mathbf{W}$:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{e}$$

Then $p(\mathbf{y})$ is Gaussian with mean vector and covariance matrix:

$$\mathbf{m}_3 = \mathbf{W}\mathbf{E}[\mathbf{x}] + \mathbf{E}[\mathbf{e}] = \mathbf{0}$$
$$\mathbf{V}_3 = \mathbf{W}\mathbf{V}_1\mathbf{W}^\top + \mathbf{V}_2$$

2.5 Product of Gaussian Densities

Let $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \mathbf{m}_1, \mathbf{V}_1)$ and $q(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \mathbf{m}_2, \mathbf{V}_2)$. Then $t(\mathbf{x}) \propto p(\mathbf{x})q(\mathbf{x})$ is Gaussian $\mathcal{N}(\mathbf{x} \mid \mathbf{m}_3, \mathbf{V}_3)$ where:

$$\mathbf{V}_3 = \left(\mathbf{V}_1^{-1} + \mathbf{V}_2^{-1}\right)^{-1}$$
$$\mathbf{m}_3 = \mathbf{V}_3\left(\mathbf{m}_1^\top \mathbf{V}_1^{-1} + \mathbf{m}_2^\top \mathbf{V}_2^{-1}\right)^\top$$

2.6 Bayesian Linear Regression

We choose the prior for $\mathbf{w}$ to be a zero-mean isotropic Gaussian:

$$p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w} \mid \mathbf{0}, \lambda^{-1}\mathbf{I}\right) \propto \exp\left\{-\frac{1}{2}\mathbf{w}^\top \lambda \mathbf{I} \mathbf{w}\right\}$$

The posterior is then Gaussian:

$$p\left(\mathbf{w} \mid \mathbf{y}, \widetilde{\mathbf{X}}, \sigma^2\right) \propto p(\mathbf{y} \mid \widetilde{\mathbf{X}}, \mathbf{w}) p(\mathbf{w})$$

$p\left(\mathbf{w} \mid \mathbf{y}, \widetilde{\mathbf{X}}, \sigma^2\right) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}, \mathbf{V})$ where:

$$\mathbf{V} = \left(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}} \sigma^{-2} + \lambda \mathbf{I}\right)^{-1}$$
$$\mathbf{m} = \mathbf{V}\sigma^{-2} \widetilde{\mathbf{X}}^\top \mathbf{y}$$

2.7 The Bayesian Predictive Distribution

The predictive distribution for the $y_\star$ of a given new corresponding $\widetilde{\mathbf{x}}_\star$ is:

$$p\left(y_\star \mid \widetilde{\mathbf{x}}_\star, \mathbf{y}, \widetilde{\mathbf{X}}\right) =$$
$$\int \mathcal{N}\left(y_\star \mid \mathbf{w}^\top \widetilde{\mathbf{x}}_\star, \sigma^2\right) \mathcal{N}(\mathbf{w} \mid \mathbf{m}, \mathbf{V}) d\mathbf{w}$$

We have that $y_\star = \mathbf{w}^\top \widetilde{\mathbf{x}}_\star + e_\star$, where $\mathbf{w} \sim \mathcal{N}(\mathbf{m}, \mathbf{V})$ and $e_\star \sim \mathcal{N}\left(0, \sigma^2\right)$. Thus:

$$p\left(y_\star \mid \widetilde{\mathbf{x}}_\star, \mathbf{y}, \widetilde{\mathbf{X}}\right) = \mathcal{N}\left(y_\star \mid m_\star, v_\star\right)$$

$m_\star = \mathbf{m}^\top \widetilde{\mathbf{x}}_\star$ and $v_\star = \widetilde{\mathbf{x}}_\star^\top \mathbf{V} \widetilde{\mathbf{x}}_\star + \sigma^2$.

2.8 MAP Inference

Maximum a posteriori (MAP) inference is a form of *regularized* MLE.

$$\mathbf{w}_{\text{MAP}} = \underset{\mathbf{w}}{\arg\max} \left\{\log p(\mathbf{y} \mid \mathbf{w}, \widetilde{\mathbf{X}}) + \log p(\mathbf{w})\right\}$$
$$= \underset{\mathbf{w}}{\arg\max} \left\{\log p(\mathbf{y} \mid \mathbf{w}, \widetilde{\mathbf{X}}) - \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w}\right\}$$
$$= \left(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}} \sigma^{-2} + \lambda \mathbf{I}\right)^{-1} \sigma^{-2} \widetilde{\mathbf{X}}^\top \mathbf{y}$$

Assumes that the posterior is well approximated by a point mass at its mode:

$$p\left(y_\star \mid \widetilde{\mathbf{x}}_\star, \mathbf{y}, \widetilde{\mathbf{X}}\right)$$
$$\approx \int p\left(y_\star \mid \widetilde{\mathbf{x}}_\star, \mathbf{w}\right) \delta\left(\mathbf{w} - \mathbf{w}_{\text{MAP}}\right) d\mathbf{w}$$
$$\approx p\left(y_\star \mid \widetilde{\mathbf{x}}_\star, \mathbf{w}_{\text{MAP}}\right)$$

MAP inference fails to generate *confidence bands* in the resulting predictions

3 Classification

3.1 Linear Classification

Given a *training set* $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ of inputs $\mathbf{x}_n = \left(x_{n,1}, \ldots, x_{n,D}\right)^\top$ and *discrete* outputs $y_n \in \{1, \ldots, C\}$, where C is the number of *classes* or *categories*, we aim to identify:

1. A partition of the input space into C *decision regions*, one for each class.
2. A measure of *confidence* (probability) in the decisions.

The decision regions are separated by *decision boundaries* at which two classes have equal predictive probability.
Deterministic linear classification maps the output of the linear model into *discrete class labels*. Assume $y_n \in \{0, 1\}$ (binary classification). Then, we can define $y_n = H\left(\mathbf{w}^\top \widetilde{\mathbf{x}}\right)$ where $H(x)$ is the Heaviside step function.

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Probabilistic linear classification maps the output of the linear model into *class probabilities*.

$$p\left(y_n = 1 \mid \widetilde{\mathbf{x}}, \mathbf{w}\right) = \sigma\left(\mathbf{w}^\top \widetilde{\mathbf{x}}\right)$$

$\sigma(\cdot)$ is a monotonically increasing function that maps $\mathbb{R}$ into $[0,1]$:

1. *The logistic function:*

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

2. *The probit function (Gaussian CDF):*

$$\sigma(x) = \int_{-\infty}^x \mathcal{N}(z \mid 0, 1) dz$$

3.2 Logistic Regression

Assume $\sigma(x)$ is the logistic function and that $y_n \in \{-1, 1\}$. Then

$$p\left(y_n \mid \mathbf{x}_n, \mathbf{w}\right) = \sigma\left(y_n \mathbf{w}^\top \widetilde{\mathbf{x}}_n\right)$$

For $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$, the log-likelihood is:

$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^N \log \sigma\left(y_n \mathbf{w}^\top \widetilde{\mathbf{x}}_n\right)$$

We can then use $d\sigma(x)/dx = \sigma(x)(1 - \sigma(x))$ to obtain the gradient:

$$\frac{d\mathcal{L}(\mathbf{w})}{d\mathbf{w}} = \sum_{n=1}^N y_n \left(1 - \sigma\left(y_n \mathbf{w}^\top \widetilde{\mathbf{x}}_n\right)\right) \widetilde{\mathbf{x}}_n$$

3.3 Gradient Ascent

The batch gradient ascent rule to maximize $\mathcal{L}(\mathbf{w})$ is:

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} + \alpha \frac{d\mathcal{L}(\mathbf{w})}{d\mathbf{w}}$$

The *learning rate* is $\alpha > 0$.
Multi-class linear classification: The *softmax function* maps the outputs into class probabilities:

$$p\left(y_n = k \mid \mathbf{w}_1, \ldots, \mathbf{w}_K, \widetilde{\mathbf{x}}_n\right)$$
$$= \frac{\exp\left(\mathbf{w}_k^\top \widetilde{\mathbf{x}}_n\right)}{\sum_{k'=1}^K \exp\left(\mathbf{w}_{k'}^\top \widetilde{\mathbf{x}}_n\right)}$$

This is equivalent to logistic regression when $C = 2$.
Non-linear logistic regression: Replace $\mathbf{x}$ with non-linear functions of the inputs $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^\top$.

4 Dimensionality Reduction

4.1 Principal Component Analysis

The *principal component analysis* (PCA) is a *linear* dimensionality reduction method. It assumes data manifold to be linear and finds the projection that minimised the squared reconstruction error.
Given a dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, with $\mathbf{x}_n \in \mathbb{R}^D$, $\sum_{n=1}^N \mathbf{x}_n = \mathbf{0}$.
By using the orthonormal basis $\{\mathbf{u}_i\}_{i=1}^D$:

$$\mathbf{x}_n = \underbrace{\sum_{i=1}^M \mathbf{x}_n^\top \mathbf{u}_i \mathbf{u}_i}_{\mathbf{x}_n'} + \underbrace{\sum_{i=M+1}^D \mathbf{x}_n^\top \mathbf{u}_i \mathbf{u}_i}_{\epsilon_n}$$

$\mathbf{x}_n'$ is the projected vector and ϵ_n is the reconstruction error. $\mathbf{u}_1, \ldots, \mathbf{u}_D$ are called the *principal component vectors*. $\mathbf{x}_n^\top \mathbf{u}_1, \ldots, \mathbf{x}_n^\top \mathbf{u}_D, n = 1, \ldots, N$, are called the *principal component scores*.
PCA finds $\mathbf{u}_1, \ldots, \mathbf{u}_D$ by minimising the

sum of square errors:

$$\text{cost}\left(\{\mathbf{u}_i\}_{i=1}^{D}\right) = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\epsilon}_n^\top \boldsymbol{\epsilon}_n = \sum_{i=M+1}^{D} \mathbf{u}_i^\top \hat{\mathbf{S}} \mathbf{u}_i$$

$\hat{\mathbf{S}} = N^{-1}\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top$ is the covariance matrix for the data.

4.2 Lagrange Multipliers

The method of *Lagrange Multipliers* allows us to solve optimization problems with equality constraints.

The Lagrangian function: Maximize $f(x,y)$ subject to $g(x,y) = c$:

$$\mathcal{L}(x,y,\lambda) = f(x,y) + \lambda(g(x,y) - c)$$

λ is called the Lagrange multiplier.

For PCA, we use Lagrange multipliers to guarantee that the $\{\mathbf{u}_i\}_{i=1}^{D}$ are normalized, i.e., $\mathbf{u}_i^\top \mathbf{u}_i = 1$ for $i = 1,\ldots,D$. The resulting Lagrangian is:

$$\mathcal{L}\left(\{\mathbf{u}_i\}_{i=M+1}^{D}, \lambda_{M+1},\ldots,\lambda_D\right) = \sum_{i=M+1}^{D}\left[\mathbf{u}_i^\top \hat{\mathbf{S}}\mathbf{u}_i + \lambda_i\left(1 - \mathbf{u}_i^\top \mathbf{u}_i\right)\right]$$

Hence $\hat{\mathbf{S}}\mathbf{u}_i = \lambda_i \mathbf{u}_i$. The $\mathbf{u}_i$ and λ_i are eigenvectors and eigenvalues of $\hat{\mathbf{S}}$.

$$\text{cost}\left(\{\mathbf{u}_i\}_{i=1}^{D}\right) = \sum_{i=M+1}^{D} \lambda_i$$

The PCA solution is given by the $\{\mathbf{u}_i\}_{i=1}^{D}$ such that:

1. $\mathbf{u}_{M+1},\ldots,\mathbf{u}_D$ are eigenvectors of $\hat{\mathbf{S}}$ with the $D-M$ smallest eigenvalues.
2. $\mathbf{u}_1,\ldots,\mathbf{u}_M$ are eigenvectors of $\hat{\mathbf{S}}$ with the M largest eigenvalues.

5 Clustering

5.1 K-means Clustering

Let $s_{n,k} = 1$ if data point n is assigned to cluster k and zero otherwise. Note that $\sum_{k=1}^{K} s_{n,k} = 1$.

K-means tries to minimise the cost function $\mathcal{C}$ with respect to $\{s_{n,k}\}$ and $\{\boldsymbol{m}_k\}$, subject to $\sum_k s_{n,k} = 1$ and $s_{n,k} \in \{0,1\}$.

$$\mathcal{C}\left(\{s_{n,k}\},\{\boldsymbol{m}_k\}\right) = \sum_{n=1}^{N}\sum_{k=1}^{K} s_{n,k}\|\boldsymbol{x}_n - \boldsymbol{m}_k\|^2$$

K-means sequentially:

1. Minimises $\mathcal{C}$ with respect to $\{s_{n,k}\}$, holding $\{\boldsymbol{m}_k\}$ fixed.
$$s_n = \arg\min_k \|\boldsymbol{x}_n - \boldsymbol{m}_k\|$$
2. Minimises $\mathcal{C}$ with respect to $\{\boldsymbol{m}_k\}$, holding $\{s_{n,k}\}$ fixed.
$$\boldsymbol{m}_k = \text{mean}\left(\boldsymbol{x}_n : s_n = k\right)$$

5.2 Mixture of Gaussians

For each data point $1\ldots N$ sample cluster membership: $p(s_n = k \mid \theta) = \pi_k$ (Note $\sum_{k=1}^{K}\pi_k = 1$) and sample datavalue given cluster membership: $p(\boldsymbol{x}_n \mid s_n = k, \theta) = \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{m}_k, \Sigma_k)$.

5.3 Kullback-Leibler Divergence

The KL-divergence is given by:

$$\mathcal{KL}(p_1(z)\|p_2(z)) = \sum_z p_1(z)\log\frac{p_1(z)}{p_2(z)}$$

The properties of KL-divergence:

1. *Gibb's inequality (non-negativity):* $\mathcal{KL}(p_1(z)\|p_2(z)) \geq 0$, equality at $p_1(z) = p_2(z)$.
2. *Non-symmetric:* $\mathcal{KL}(p_1(z)\|p_2(z)) \neq \mathcal{KL}(p_2(z)\|p_1(z))$.

We can obtain a lower bound on the log-likelihood $\log p(\boldsymbol{x}\mid\theta)$ using an arbitrary distribution over class memberships $q(\boldsymbol{s})$:

$$\mathcal{F}(q(\boldsymbol{s}),\theta) = \log p(\boldsymbol{x}\mid\theta) - \sum_{\boldsymbol{s}} q(\boldsymbol{s})\log\frac{q(\boldsymbol{s})}{p(\boldsymbol{s}\mid\boldsymbol{x},\theta)}$$

$\mathcal{F}(q(\boldsymbol{s}),\theta)$ is called the free-energy which equals to log-likelihood when $q(\boldsymbol{s}) = p(\boldsymbol{s}\mid\boldsymbol{x},\theta)$.

$$\mathcal{F}(q(\boldsymbol{s}),\theta) = \sum_{\boldsymbol{s}} q(\boldsymbol{s})\log\frac{p(\boldsymbol{x}\mid\boldsymbol{s},\theta)p(\boldsymbol{s}\mid\theta)}{q(\boldsymbol{s})}$$

5.4 The Expectation Maximisation (EM) Algorithm

From initial (random) parameters θ_0 iterate $t = 1,\ldots,T$ the two steps:

1. *E step*: For fixed θ_{t-1}, maximize lower bound $\mathcal{F}(q(\boldsymbol{s}),\theta_{t-1})$ wrt $q(\boldsymbol{s})$. As log likelihood $\log p(\boldsymbol{x}\mid\theta)$ is independent of $q(\boldsymbol{s})$ this is equivalent to minimizing $\mathcal{KL}(q(\boldsymbol{s})\|p(\boldsymbol{s}\mid\boldsymbol{x},\theta_{t-1}))$, so $q_t(\boldsymbol{s}) = p(\boldsymbol{s}\mid\boldsymbol{x},\theta_{t-1})$.
2. *M step*: For fixed $q_t(\boldsymbol{s})$ maximize the lower bound $\mathcal{F}(q_t(\boldsymbol{s}),\theta)$ wrt θ.

$$\theta_t = \underset{\theta}{\text{argmax}} \sum_{\boldsymbol{s}} q_t(\boldsymbol{s})\log(p(\boldsymbol{x}\mid\boldsymbol{s},\theta)p(\boldsymbol{s}\mid\theta))$$

Mixture of Gaussians: Probability of the observations given the latent variables and the parameters, and the prior on latent variables are:

$$p(x_n \mid s_n = k, \theta) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{1}{2\sigma_k^2}(x_n - \mu_k)^2}$$

$$p(s_n = k \mid \theta) = \pi_k$$

The E step becomes:

$$q(s_n = k) \propto \frac{\pi_k}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{1}{2\sigma_k^2}(x_n - \mu_k)^2} = u_{nk}$$

$$q(s_n = k) = r_{nk} = \frac{u_{nk}}{\sum_{k=1}^{K} u_{nk}}$$

r_{nk} is called the *responsibility* that component k takes for data point n.

The M step, optimizing $\mathcal{F}(q(s),\theta)$ wrt the parameters, θ:

$$\mu_j = \frac{\sum_{n=1}^{N} q(s_n = j)x_n}{\sum_{n=1}^{N} q(s_n = j)}$$

$$\sigma_j^2 = \frac{\sum_{n=1}^{N} q(s_n = j)\left(x_n - \mu_j\right)^2}{\sum_{n=1}^{N} q(s_n = j)}$$

$$\pi_j = \frac{1}{N}\sum_{n=1}^{N} q(s_n = j)$$

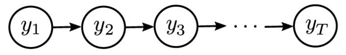
6 Sequence Modelling

6.1 Markov Models for Discrete Data

Markov models of order n are called n-gram models.

First order Markov (bi-gram):

$$p(y_1, y_2, \ldots, y_T) = p(y_1)p(y_2 \mid y_1)\ldots p(y_T \mid y_{T-1})$$

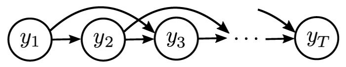


$$\begin{tikzcd} y_1 \arrow[r] & y_2 \arrow[r] & y_3 \arrow[r] & \cdots \arrow[r] & y_T \end{tikzcd}$$

Discrete states $y_t \in \{1,\ldots,K\}$ with initial state probabilities $p(y_1 = k) = \pi_k^0$ and transition probabilities (stochastic matrix) $p(y_t = k \mid y_{t-1} = l) = T_{k,l}$ $(\sum_{k=1}^{K} T_{k,l} = 1)$.

Second order Markov (tri-gram):

$$p(y_1, y_2, \ldots, y_T) = p(y_1)p(y_2 \mid y_1)\ldots p(y_T \mid y_{T-1}, y_{T-2})$$



$$\begin{tikzcd} y_1 \arrow[r] \arrow[rr, bend left] & y_2 \arrow[r] \arrow[rr, bend left] & y_3 \arrow[r] & \cdots \arrow[r] \arrow[r, bend left] & y_T \end{tikzcd}$$

The transition probabilities $p(y_t = k \mid y_{t-1} = l, y_{t-2} = m) = T_{k,l,m}$. Hence n-grams require large multidimensional arrays.

6.2 Markov Models for Continuous Data

$AR(q)$ is the *Auto-Regressive* (AR) Gaussian model of order q.

First order Markov ($AR(1)$):

Continuous vector states $y_t \in \mathbb{R}^D$ with initial state density $p(y_1) = \mathcal{G}(y_1; \mu_0, \Sigma_0)$ and transition density $p(y_t \mid y_{t-1}) = \mathcal{G}(y_t; \Lambda y_{t-1}, \Sigma)$.

Second order Markov ($AR(2)$):

The transition density is $p(y_t \mid y_{t-1}, y_{t-2}) = \mathcal{G}(y_t; \Lambda_1 y_{t-1} + \Lambda_2 y_{t-2}, \Sigma)$. Joint distribution over all variables is always multivariate Gaussian.

7 Hidden Markov Models

Discrete Hidden State $x_t \in \{1,\ldots,K\}$:

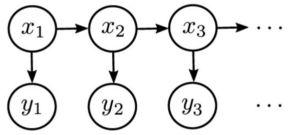
$$p(x_t = k \mid x_{t-1} = l) = T_{k,l}$$

Continuous Observed State:

$$p(y_t \mid x_t = k) = \mathcal{G}(y_t; \mu_k, \Sigma_k)$$

Discrete Observed State:

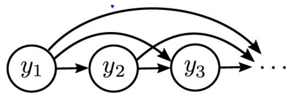
$$p(y_t = l \mid x_t = k) = S_{l,k}$$



$$\begin{tikzcd} x_1 \arrow[r] \arrow[d] & x_2 \arrow[r] \arrow[d] & x_3 \arrow[r] \arrow[d] & \cdots \\ y_1 & y_2 & y_3 & \cdots \end{tikzcd}$$

$$p(y_{1:T}, x_{1:T}) = \prod_{t=1}^{T} p(x_t \mid x_{t-1})p(y_t \mid x_t)$$

Marginalise latents to obtain fully connected model:



$$\begin{tikzcd} y_1 \arrow[r] \arrow[rr, bend left] \arrow[rrr, bend left] & y_2 \arrow[r] \arrow[rr, bend left] & y_3 \arrow[r] & \cdots \end{tikzcd}$$

7.1 Linear Gaussian State Space Models (LGSSMs)

Continuous Hidden State $x_t \in \mathbb{R}^K$:

$$p(x_t \mid x_{t-1}) = \mathcal{G}(x_t; Ax_{t-1}, Q)$$

Continuous Observed State $y_t \in \mathbb{R}^D$:

$$p(y_t \mid x_t) = \mathcal{G}(y_t; Cx_t, R)$$

7.2 Inference

Distributional estimates:

	marginal	joint
filter	$p(x_t \mid y_{1:t})$	$p(x_{1:t} \mid y_{1:t})$
smoother	$p(x_t \mid y_{1:T})$	$p(x_{1:T} \mid y_{1:T})$

Point estimates:

1. Most probable state at t:
$$x_t^* = \arg\max_{x_t} p(x_t \mid y_{1:T})$$
2. Most probable sequence:
$$x_{1:T}' = \arg\max_{x_{1:T}} p(x_{1:T} \mid y_{1:T})$$

Kalman Filter:

Given $p(x_{t-1} \mid y_{1:t-1}) = \mathcal{G}\left(x_{t-1}; \mu_{t-1}^{t-1}, V_{t-1}^{t-1}\right)$ where the superscript is the most recent data used in prediction and the subscript is the variable being predicted. Diffuse via dynamics:

$$p(x_t \mid y_{1:t-1}) = \int p(x_t \mid x_{t-1})\, p(x_{t-1} \mid y_{1:t-1})\, \mathrm{d}x_{t-1}$$

Since $x_t = Ax_t + Q^{1/2}\varepsilon_t$, we obtain $p(x_t \mid y_{1:t-1}) = \mathcal{G}\left(x_t; \mu_t^{t-1}, V_t^{t-1}\right)$:

$$\begin{aligned} \mu_t^{t-1} &= A\mu_{t-1}^{t-1} \\ V_t^{t-1} &= AV_{t-1}^{t-1}A^\top + Q \end{aligned}$$

Mean diffuses toward 0 and variance inflates. Combine with likelihood:

$$\begin{aligned} p(x_t \mid y_{1:t}) &\propto p(x_t \mid y_{1:t-1})\, p(y_t \mid x_t) \\ &= \mathcal{G}\left(x_t; \mu_t^t, V_t^t\right) \end{aligned}$$

Defining the Kalman gain $K_t = V_t^{t-1}C^\top\left(CV_t^{t-1}C^\top + R\right)^{-1}$:

$$\begin{aligned} \mu_t^t &= \mu_t^{t-1} + K_t\left(y_t - C\mu_t^{t-1}\right) \\ V_t^t &= V_t^{t-1} - K_tCV_t^{t-1} \end{aligned}$$

Forward Algorithm:

Given $p(x_{t-1} = k \mid y_{1:t-1}) = \rho_{t-1}^{t-1}(k)$. Diffuse via dynamics:

$$p(x_t = k \mid y_{1:t-1}) = \sum_{l=1}^{K} p(x_t = k \mid x_{t-1} = l)\, p(x_{t-1} = l \mid y_{1:t-1})$$

We obtain $\rho_t^{t-1}(k) = \sum_{l=1}^{K} T(k,l)\rho_{t-1}^{t-1}(l)$. Combine with likelihood:

$$p(x_t = k \mid y_{1:t}) \propto p(x_t = k \mid y_{1:t-1})\, p(y_t \mid x_t = k)$$

Hence $\rho_t^t(k) \propto \rho_t^{t-1}(k) p(y_t \mid x_t = k)$.

(The End)